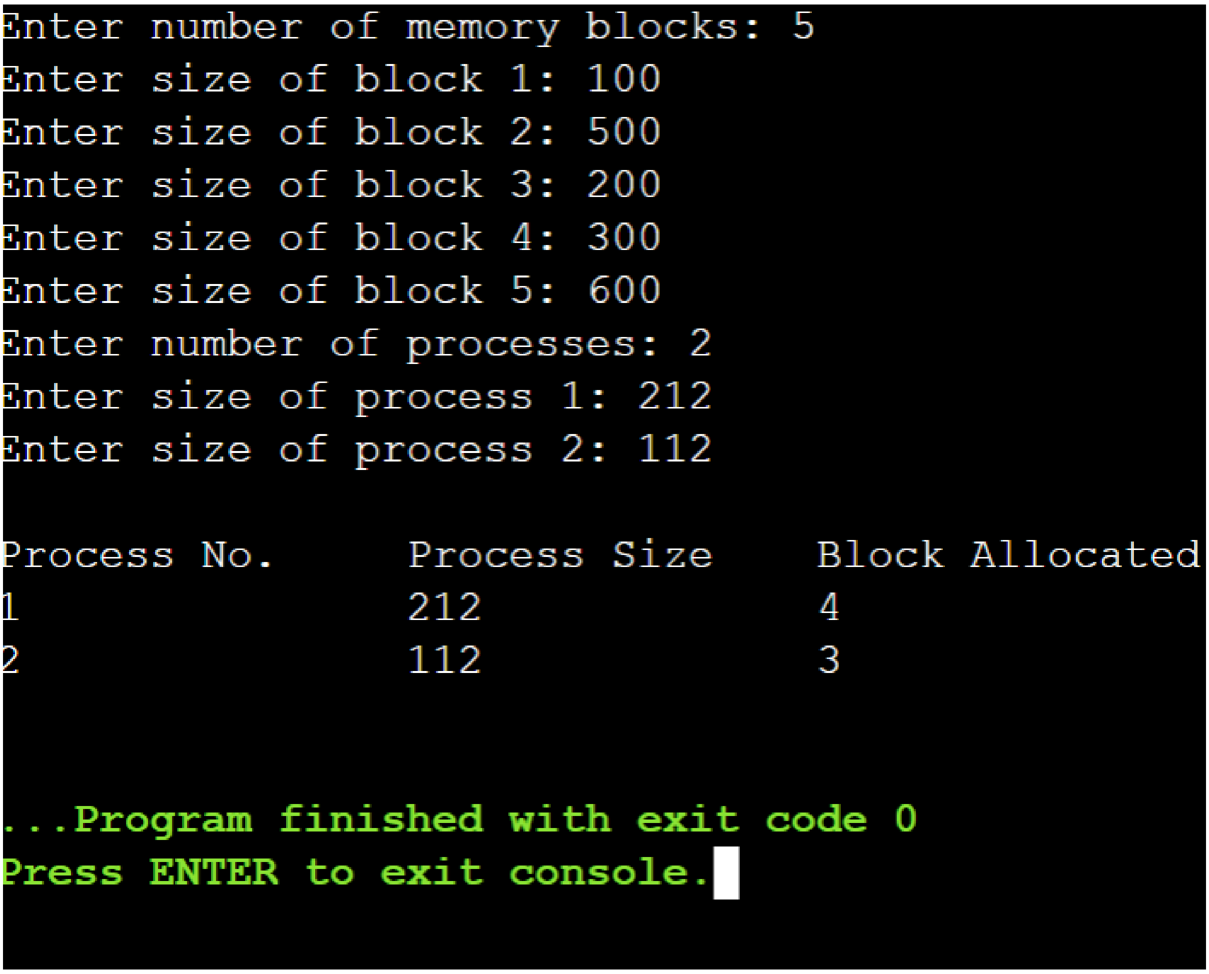
```
#include <stdio.h>

int main() {
    int n, m;
    printf("Enter number of memory blocks: ");
    scanf("%d", &n);

    int blockSize[n], blockAllocated[n];
    for (int i = 0; i < n; i++) {
        printf("Enter size of block %d: ", i + 1);
        scanf("%d", &blockSize[i]);
        blockAllocated[i] = 0; // 0 means free
    }

    printf("Enter number of processes: ");
    scanf("%d", &m);

    int processSize[m], allocation[m];
    for (int i = 0; i < m; i++) {
        printf("Enter size of process %d: ", i + 1);
        scanf("%d", &processSize[i]);
        allocation[i] = -1; // -1 means not allocated
    }

    // Best Fit Allocation
    for (int i = 0; i < m; i++) {
        int bestIndex = -1;
        for (int j = 0; j < n; j++) {
            if (!blockAllocated[j] && blockSize[j] >= processSize[i]) {
                if (bestIndex == -1 || blockSize[j] < blockSize[bestIndex]) {
                    bestIndex = j;
                }
            }
        }

        if (bestIndex != -1) {
            allocation[i] = bestIndex;
            blockAllocated[bestIndex] = 1; // mark block as allocated
        }
    }

    // Display allocation results
    printf("\nProcess No.\tProcess Size\tBlock Allocated\n");
    for (int i = 0; i < m; i++) {
        printf("%d\t\t%d\t\t", i + 1, processSize[i]);
        if (allocation[i] != -1)
            printf("%d\n", allocation[i] + 1);
        else
            printf("Not Allocated\n");
    }

    return 0;
}
```



```
Enter number of memory blocks: 5
Enter size of block 1: 100
Enter size of block 2: 500
Enter size of block 3: 200
Enter size of block 4: 300
Enter size of block 5: 600
Enter number of processes: 2
Enter size of process 1: 212
Enter size of process 2: 112

Process No.     Process Size    Block Allocated
1               212             4
2               112             3


...Program finished with exit code 0
Press ENTER to exit console.
```